Parsing Japanese with a PCFG treebank grammar

Tsaiwei FANG* Alastair BUTLER†‡ Kei YOSHIMOTO*‡
*Graduate School of International Cultural Studies, Tohoku University
†PRESTO, Japan Science and Technology Agency
‡Center for the Advancement of Higher Education, Tohoku University

Abstract

This paper describes constituent parsing of Japanese using a probabilistic context-free grammar treebank grammar that is enhanced with Parent Encoding, reversible tree transformations, refinement of treebank labels and Markovisation. We evaluate the quality of the resulting parsing.

1 Introduction

The focus of this paper is on the task of obtaining a constituent parser for Japanese using a treebank (Keyaki Treebank) as training data and a Probabilistic Context-Free Grammar (PCFG) as parsing method. We report results for a vanilla PCFG model that is directly read off the training data and for an enhanced PCFG model obtained with transformations of the training data that aim to tune the treebank representations to the specific needs of probabilistic context-free parsers, while allowing for the original annotation to be restored. Specifically, we follow best practices from Johnson (1998), Klein and Manning (2003) and Fraser et al. (2013) among others of Parent Encoding, reversible tree transformations, refinement of treebank labels and Markovisation. PCFGs so derived can be used by a parser to construct maximal probability (Viterbi) parses. We evaluate the quality of the resulting parsing using standard PARSEVAL constituency measures. Our fully labelled bracketing score for a held-out portion of the Keyaki Treebank (1,300 trees) is 79.97 (recall), 80.61 (precision) and 80.29 (F-score). We show a learning curve suggestive that parser performance will continue to strongly improve with access to more training data.

Table 1: Keyaki Treebank content

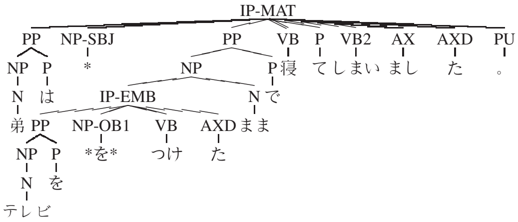
Domain	Number of trees
blog posts	217
Japanese Law	484
newspaper	1600
telephone calls	1177
textbooks	7733
Wikipedia	2464
Total	13675

2 The parser

The parsing of this paper is made possible because of the unlexicalised statistical parser BitPar (Schmid, 2004), which allows any grammar rule files in the proper format to be used for parsing. BitPar uses a fast bitvector-based implementation of the Cocke-Younger-Kasami algorithm, storing the parse chart as a large bit vector. This enables full parsing (without search space pruning) with large treebank grammars. BitPar can extract from the parse chart the most likely parse tree (Viterbi parse), or the full set of parses in the form of a parse forest, or the n-best parse trees.

3 The treebank

The grammar and lexicon used by the BitPar parser are extracted from the Keyaki Treebank (Butler et al. 2012). The current composition of the Keyaki Treebank is detailed in Table 1. The treebank uses an annotation scheme that follows, with adaptations for Japanese, the general scheme proposed in the *Annotation manual for the Penn Historical Corpora and the PCEEC* (Santorini 2010). Constituent structure is represented with labelled bracketing and augmented with grammatical functions and notation for recovering discontinuous constituents. Primary motivation for the annotation has been to facilitate automated searches for linguistic research (e.g., via CorpusSearch[1]), and to provide a syntactic base that is sufficiently rich to enable an automatic generation of (higher-order) predicate logic based meaning representations.[2] A typical parse in tree form looks like:



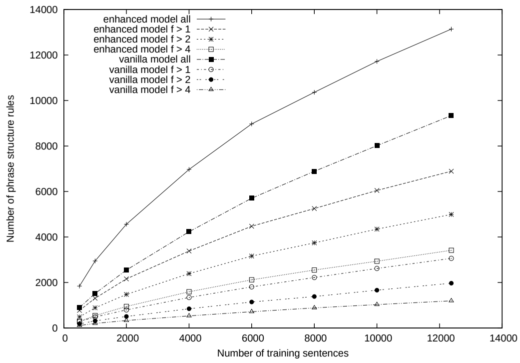
Every word has a word level part-of-speech label. Phrasal nodes (NP, PP, ADJP, etc.) immediately dominate the phrase head (N, P, ADJ, etc.), so that the phrase head has as sisters both modifiers and complements. Modifiers and complements are distinguished because there are extended phrase labels to mark function (e.g., -EMB encodes that the clause テレビをつけた is a complement of the phrase head まま). All noun phrases immediately dominated by IP are marked for function (NP-SBJ=subject, NP-OB1=direct object, NP-TMP=temporal NP, etc.). The PP label is never extended with function marking. However the immediately following sibling of a PP may be present in the annotation to provide disambiguation information for the PP. Thus, `(NP-OB1 *を*)` indicates the immediately preceeding PP (with case particle を) is the object, while `(NP-SBJ *)` indicates the immediately preceeding PP (without case particle) is the subject. All clauses have extended labels to mark function (IP-MAT=matrix clause, IP-ADV=adverbial clause, IP-REL=relative clause, etc.).

4 Extracted grammars

Figure 1 shows the growth of extracted phrase structure rules for a vanilla grammar model directly read off the Keyaki Treebank and also for an enhanced model. The enhanced model is obtained after reversible changes are made to the treebank aimed at improving parsing quality. This section focuses on the changes made for the enhanced model. Specifically

[1] http://corpussearch.sourceforge.net
[2] http://www.compling.jp/ts

Figure 1: growth of phrase structure rules



this was created with techniques from Johnson (1998), Klein and Manning (2003) and Fraser et al. (2013) among others, of:

1. eliminating discontinuous constituents and (Section 4.1),
2. transforming and augmenting treebank annotations (Section 4.2), and
3. following the extraction of phrase structure rules, lexical rules, and their frequencies from the annotated parse trees, markovising the grammar (Section 4.3).

Note that all curves of Figure 1 remain steep at the maximum training set size of 12,375 trees, suggesting more data would lead to more significant growth. As a comparison, the treebank grammar of Schmid (2006) extracted from the Penn Treebank of English has 52,297 phrase structure rules (enabling a labelled bracketing F-score of 86.6%).

4.1 Discontinuous constituents

The Keyaki Treebank annotates trace nodes, for example with relative clauses. But unlike the Penn Treebank (Bies et al. 1995) trace nodes are not indexed and typically appear clause initially, with precise attachment points unspecified since it is enough to assume that constituents at the IP level are dependents of the main verb of the clause. For trace nodes within embedded contexts (cases of long distance dependency) it is the phrase level of attachment for the trace node that is the relevant indicator of the dependency. For the current work we assume parse trees from which trace nodes are removed and aim to recognize discontinuous constituents in a post-processing step, following for example Johnson (2001).

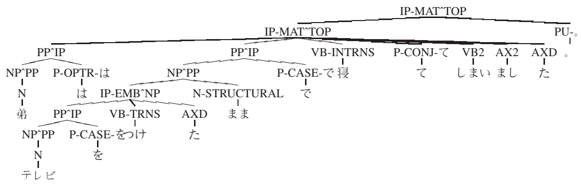
4.2 Transforming and augmenting annotations

In addition to removing trace nodes, transformations and augmentations of the trees are performed. Specifically interjection, punctuation or parenthetical materials occurring at the left or right periphery of a constituent are moved to a new projection of the constituent. There is also Parent Encoding, following Johnson (1998), which copies the syntactic label of a parent node (minus the functional information) onto the labels of its children. Finally there are refinements to the POS tags.

Refinements to the POS tags include the P (particle) tag becoming either P-CASE, P-CONJ, P-COORD, P-CP-THT, P-FINAL, P-NOUN or P-OPTR depending on the functional role of the particle and/or the syntactic context in which the particle occurs. Verbs are also split to inform information about the clause of occurrence: VB-THT (verb with CP-THT (clausal) complement), VB-DITRNS (triggered by NP-OB2), VB-TRNS (triggered by NP-OB1), VB-INTRNS (default encoding).

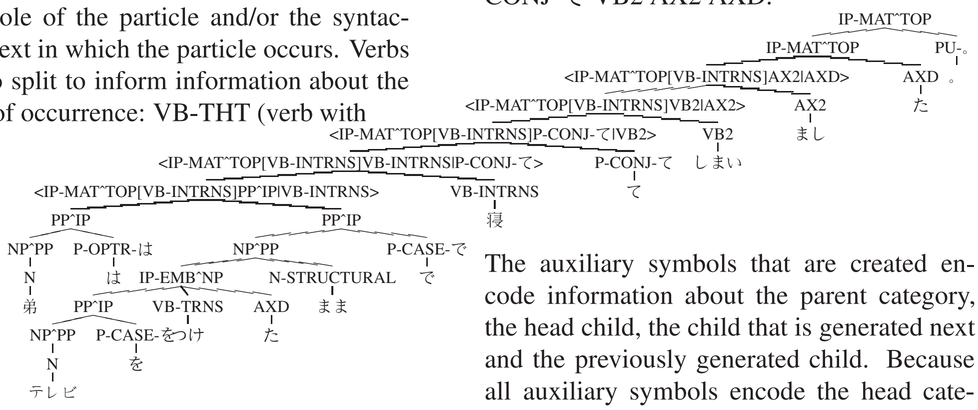
In addition, the POS tags of the most frequent particles, の, は, に, を, が, て, と, で, も, か, から, etc., are marked with a feature to identify the specific particle. For example, と can be tagged as either P-CASE-と, P-CONJ-と, P-COORD-と or P-CP-THT-と. This can be seen as a restricted form of lexicalisation. In the same way, the auxiliary verbs あっ, あり, ある, あれ, あろ, で and な are "lexicalised", as is the negation marker ず and certain punctuations, e.g., '。'. However, other similar enrichments of the POS information was found to hurt overall parsing performance.

Applying the above changes to the tree of section 3 results in the following tree representation:



4.3 Markovisation

The Keyaki Treebank uses rather flat structures, particularly at the clause level, with nodes having up to 31 child nodes. As Fraser et al. (2013) note this causes problems because only some rules of that length appear in the training data. This sparse data problem is solved by markovisation (Collins 1997), which splits long rules into a set of shorter rules. The following shows consequences to our running example of markovising the rule IP-MAT^TOP –> PP^IP PP^IP VB-INTRNS P-CONJ-て VB2 AX2 AXD.



The auxiliary symbols that are created encode information about the parent category, the head child, the child that is generated next and the previously generated child. Because all auxiliary symbols encode the head category, the head is selected by the first rule, while being generated later. The markovisation strategy is currently set to transform rules that occur less than 60 times in the training data.

5 Parsing experiments

The treebank was randomised because of the diverse nature of the treebank (see Table 1) and a dataset of 1,300 trees was held-out for testing and the remaining 12,375 trees were used for training. In all evaluations, we used the original Keyaki parse trees as gold standard and converted the parse trees generated by our parser to the same format by undoing the transformations and removing the augmentations. Perfect segmentation was used for all

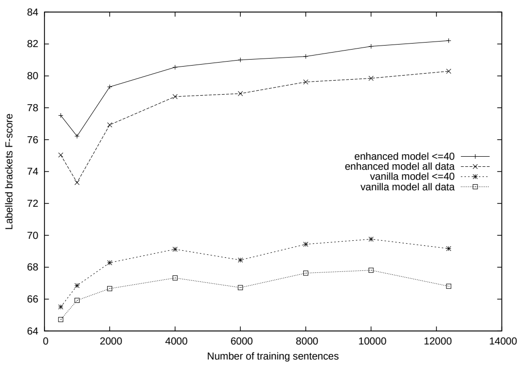
Table 2: PARSEVAL results (results for sentence lengths ≤ 40 in brackets)

Model	precision	recall	F-score
Vanilla	65.73 (68.10)	67.92 (70.27)	66.81 (69.17)
Enhanced	79.97 (81.84)	80.61 (82.58)	80.29 (82.21)

evaluations, being necessary to obtain a PARSEVAL score.

The results of parsing using the vanilla and enhanced PCFG models on the test data are given in Table 2, using the standard PARSEVAL measures (Black et al., 1991), i.e., values for bracketing precision, recall, and F-score, but for fully labelled evaluation only. Figure 2 shows learning curves for all sentences and for sentence lengths ≤ 40. In contrast to the two curves of the vanilla model, the two curves of the enhanced model remain steep at the maximum training set size of 12,375 trees. Figure 3 shows coverage results for the models with differing amounts of training data. The enhanced model offers high coverage early on, a consequence of the markovisation. This also explains some of the apparent lack of growth or even loss in F-score, as F-score is only calculated from valid parsings.

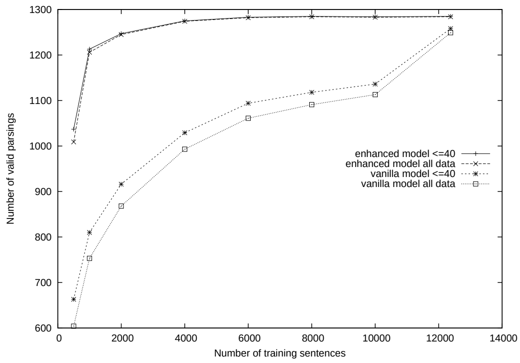
Figure 2: learning curves for all sentences and sentence lengths ≤ 40



6 Conclusion

This paper has presented a PCFG treebank grammar for Japanese trained on the Keyaki treebank. Parsing performance was enhanced with Parent Encoding, reversible tree transformations, refinement of treebank labels and Markovisation. This establishes a significant parsing baseline for Japanese, that appears to be competitive with other attempts at constituency parsing of Japanese, notably Tanaka and Nagata (2013). Our results strongly suggest that the enhanced parsing model would benefit considerably from the availability of more training data. More training data is expected to also enable improvements from a yet more fine-grained label set.

Figure 3: coverage results



References

Alastair Butler, Zhu Hong, Tomoko Hotta, Ruriko Otomo, Kei Yoshimoto and Zhen Zhou. 2012. Keyaki Treebank: phrase structure with functional information for Japanese. In *Proceedings of Text Annotation Workshop*.

Fraser, Alexander, Helmut Schmid, Richárd Farkas, Renjing Wang and Hinrich Schutze. 2013. Knowledge Sources for Constituent Parsing of German, a Morphologically Rich and Less-Configurational Language. *Computational Linguistics*, 39(1):57–85, 2013.

Johnson, Mark. 1998. PCFG models of linguistic tree representations. *Computational Linguistics*, 24(4):613–632, 1998.

Johnson, Mark. 2001. A simple pattern-matching algorithm for recovering empty nodes and their antecedents. In *ACL*, pages 136–143.

Klein, Dan and Chris Manning. 2003. Accurate unlexicalized parsing. In *ACL-03*.

Schmid, Helmut. 2006. Trace prediction and recovery with unlexicalized PCFGs and slash features. In *COLING-ACL*, pages 177–184.

Schmid, Helmut. 2004. Efficient parsing of highly ambiguous context-free grammars with bit vectors. In *COLING*, pages 162–168.

Tanaka, Takaaki and Masaaki Nagata. 2013. Constructing a Practical Constituent Parser from a Japanese Treebank with Function Labels. *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically Rich Languages*, pages 108–118.